IN THE CLAIMS:

1. (Original) A method for detecting resource exception errors comprising the steps of:

scanning a code for a first method invocation used to open a first resource filc;

identifying said first method invocation: and

opening said first resource file using said first method invocation to detect resource exception errors.

2. (Original) The method as recited in claim 1 further comprising the steps:

scanning said code for a first method signature; and

scanning said code for a first pair of string delimiters adjacent to said first method signature, wherein a string within said first pair of string delimiters adjacent to said first method signature is a key of said first resource file.

3. (Previously presented) The method as recited in claim 2, wherein said first method signature indicates said first resourcc file.

4. (Original) The method as recited in claim 2, wherein said first method signature is a first parameter of said first method invocation.

5. (Original) The method as recited in claim 2, wherein said key of said first resource file is a second parameter of said first method invocation.

6. (Original) The method as recited in claim 2 further comprising the step of:

determining whether said kcy and its associated valuc of said first resource file are defined in said first resource filc.

7. (Original) The method as recited in claim 6, wherein if said key or its associated value of said first resource file is not dcfined in said first resource file, then a resource exception error is detected.

8. (Original) The method as recited in claim 6, wherein if said key and its associated value of said first resource file are defined in said first resource file, then the method further comprises the step of:

determining whether to scan more code for a second method invocation used to open a second resource file.

9. (Original) The method as recited in claim 8, wherein if there is more code to scan, then the method further comprises the step of:

scanning said code for said second method invocation used to open said second resource file.

10. (Original) The method as recited in claim 9 further comprising the steps of:

identifying said second method invocation; and

opening said second resource file using said second method invocation to detect resource exception errors.

11. (Original) The method as recited in claim 9, wherein said second method invocation was not identified, wherein the method further comprises the step of:

generating a report.

12. (Original) The method as recited in claim 11, wherein said report comprises a listing of all resource exception errors detected.

13. (Original) The method as recited in claim 8, wherein if there is no more code to scan, then the method further comprises the step of:

generating a report.

14. (Original) The method as recited in claim 13, wherein said report comprises a listing of all resource exception errors detected.

15. (Original) The method as recited in claim 7 further comprising the step of:

determining whether to scan more code for a second method invocation used to open a second resource file.

16. (Original) The method as recited in claim 15, wherein if there is more code to scan, then the method further comprises the step of:

scanning said code for said second method invocation used to open said second resource file.

17. (Original) The method as recited in claim 16 further comprising the steps of:

identifying said second method invocation; and

opening said second resource file using said second method invocation to detect resource exception errors.

18. (Original) The method as recited in claim 16, wherein said second method invocation was not identified, wherein the method further comprises the step of:

generating a report.

19. (Original) The method as recited in claim 18, wherein said report comprises a listing of all resource exception errors detected.

20. (Original) The method as recited in claim 15, wherein if there is no more code to scan, then the method further comprises the step of:

generating a report.

21. (Original) The method as recited in claim 20, wherein said report comprises a listing of all resource exception errors detected.

22. (Original) A computer program product in a computer readable medium for detecting resource exception errors, comprising:

programming operable for scanning a code for a first method invocation used to open a first resource file;

programming operable for identifying said first method invocation; and

programming operable for opening said first resource file using said first method invocation to detect resource exception errors.

23. (Original) The computer program product as recited in claim 22 further comprises:

programming operable for scanning said code for a first method signature; and

programming operable for scanning said code for a first pair of string delimiters adjacent to said first method signature, wherein a string within said first pair of string delimiters adjacent to said first method signature is a key of said first resource file.

24. (Previously presented) The computer program product as recited in claim 23, wherein said first method signature indicates said first resource file.

25. (Original) The computer program product as recited in claim 23, wherein said first method signature is a first parameter of said first method invocation.

26. (Original) The computer program product as recited in claim 23, wherein said key of said first resource file is a second parameter of said first method invocation.

27. (Original) The computer program product as recited in claim 23 further comprises:

programming operable for determining whether said key and its associated value of said first resource file are defined in said first resource file.

28. (Original) The computer program product as recited in claim 27, wherein if said key or its associated value of said first resource file is not defined in said first resource file, then a resource exception error is detected.

29. (Original) The computer program product as recited in claim 27, wherein if said key and its associated value of said first resource file are defined in said first resource file, then the computer program product further comprises:

programming operable for determining whether to scan more code for a second method invocation used to open a second resource file.

30. (Original) The computer program product as recited in claim 29, wherein if there is more code to scan, then the computer program product further comprises:

programming operable for scanning said code for said second method invocation used to open said second resource file.

31. (Original) The computer program product as recited in claim 30 further comprises:

programming operable for identifying said second method invocation; and

programming operable for opening said second resource file using said second method invocation to detect resource exception errors.

32. (Original) The computer program product as recited in claim 30, wherein said second method invocation was not identified, wherein the computer program product further comprises:

programming operable for generating a report.

33. (Original) The computer program product as recited in claim 32, wherein said report comprises a listing of all resource exception errors detected.

34. (Original) The computer program product as recited in claim 29, wherein if there is no more code to scan, then the computer program product further comprises:

programming operable for generating a report.

35. (Original) The computer program product as recited in claim 34, wherein said report comprises a listing of all resource exception errors detected.

36. (Original) The computer program product as recited in claim 28 further comprises:

programming operable for determining whether to scan more code for a second method invocation used to open a second resource file.

37. (Original) The computer program product as recited in claim 36, wherein if there is more code to scan, then the computer program product further comprises:

programming operable for scanning said code for said second method invocation used to open said second resource file.

38. (Original) The computer program product as recited in claim 37 further comprises:

programming operable for identifying said second method invocation; and

programming operable for opening said second resource file using said second method invocation to detect resource exception errors.

39. (Original) The computer program product as recited in claim 37, wherein said second method invocation was not identified, wherein the computer program product further comprises:

programming operable for generating a report.

40. (Original) The computer program product as recited in claim 39, wherein said report comprises a listing of all resource exception errors detected.

41. (Original) The computer program product as recited in claim 36, wherein if there is no more code to scan, then the computer program product further comprises:

programming operable for generating a report.

42. (Original) The computer program product as recited in claim 41, wherein said report comprises a listing of all resource exception errors detected.

43. (Original) A data processing systcm, comprising:

a processor;

a memory unit for storing instructions of said processor;

an input mechanism;

an output mechanism;

a bus system for coupling the processor to thc memory unit, input mechanism, and output mechanism,

means for scanning a code for a first method invocation used to open a first resource file;

means for identifying said first method invocation; and

mcans for opcning said first resourcc file using said first method invocation to detect rcsource exception errors.

44. (Original) The data processing system as recited in claim 43, wherein the system further compriscs:

means for scanning said code for a first method signature; and

means for scanning said code for a first pair of string delimiters adjacent to said first method signature, wherein a string within said first pair of string delimiters adjacent to said first method signature is a key of said first resource file.

45. (Previously prescnted) The data processing system as recited in claim 44, wherein said first method signature indicates said first resource file.

46. (Original) The data processing systcm as recited in claim 44, wherein said first method signature is a first parameter of said first method invocation.

47. (Original) The data processing system as recited in claim 44, wherein said key of said first resource file is a sccond parametcr of said first method invocation.

48. (Original) The data processing systcm as recitcd in claim 44, wherein the system further comprises:

means for determining whether said key and its associated value of said first resource file are defined in said first resource file.

49. (Original) The data processing system as recited in claim 48, wherein if said key or its associated value of said first resource file is not defined in said first resource file, then a resource exception error is detected.

50. (Original) The data processing system as recited in claim 48, wherein if said key and its associated value of said first resource file are defined in said first resource file, then the system further comprises:

means for determining whether to scan more code for a second method invocation used to open a second resource file.

51. (New) The method of Claim 1 wherein the resource file is an associative array of keys and values.

52. (New) The computer program product of Claim 22, further comprising at least one additional resource file, where the first resource file comprises a plurality of text strings in a first language and the at least one additional resource file comprises a plurality of text strings in a second language that is different from the first language.

53. (New) The data processing system of Claim 43, wherein the resource file comprises locale specific text strings.